	Type	Hits	Search Text	DBs	Time Stamp	Comments	Error Definition	Errors
1	BRS	255	garbage same cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/18 15:37			0
2	BRS	102	(garbage same cache) same object	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/18 15:37			0
3	BRS	12	((garbage same cache) same object) and consecutive	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 11:27			0
4	BRS	41	"write combine"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 11:28			0
5	BRS	1	"write combine" and garbage	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 11:27			0
6	BRS	10	"write combine" and consecutive	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 11:32			0
7	BRS	8	("write combine" and consecutive) and cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 11:32			0
8	BRS	68	"moving garbage"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 11:34			0
9	BRS	18	"moving garbage" and cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:52			0
10	BRS	4612	without near4 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:52			0
11	BRS	96	"in-space"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:53			0
12	BRS	1	"out-space"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:53			0
13	BRS	9581	"out space"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:53			0
14	BRS	16	(without near4 cache) and "out space"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:55			0
15	BRS	1	"to space"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 12:56			0
16	BRS	2	"from space"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 13:04			0
17	BRS	2	"6421689"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 13:04			0
18	BRS	170	amount near4 data near4 copied	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 14:20			0
19	BRS	3377	CPU near4 parameter	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 14:20			0
20	BRS	3	(amount near4 data near4 copied) and (CPU near4 parameter)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/19 14:20			0

PALM INTRANET

Day : Saturday
Date: 7/19/2003
Time: 10:52:49

Inventor Name Search Result

Your Search was:

Last Name = HUDSON
First Name = RICHARD

Application#	Patent#	Status	Date Filed	Title	Inventor Name
60274182	Not Issued	020	03/09/2001	SOFT, NON-MARRING, DISPOSABLE, REPLACEABLE, LONG LASTING, LOW COST, UNIVERSAL VISE JAW COVER(S) AND OR VISE JAW(S). (A WAY OF MODIFYING AND IMPROVING THE PERFORMANCE AND USEFULNESS OF A MECHANICAL WORK HOLDING DEVISE	HUDSON, RICHARD CRAIG
60216771	Not Issued	020	07/07/2000	METHODS FOR PROTECTION OF STRATIFIED SQUAMOUS EPITHELIUM AGAINST INJURY BY NOXIOUS SUBSTANCES AND NOVEL AGENTS FOR USE THEREFOR	HUDSON, RICHARD A.
60062441	Not Issued	159	10/17/1997	SMART BATTERY SYSTEM AND INTERFACE	HUDSON , RICHARD A.
60062280	Not Issued	159	10/17/1997	SMART BATTERY SYSTEM AND INTERFACE	HUDSON , RICHARD A.
60062279	Not Issued	159	10/17/1997	SMART BATTERY SYSTEM AND INTERFACE	HUDSON , RICHARD A.
10411742 Nano Padmanabhan 711/163	Not Issued	030	04/10/2003	METHOD, APPARATUS AND ARTICLE FOR LOCK MANAGEMENT	HUDSON, RICHARD L.
10327557 DOU Popovich 707/206	Not Issued	030	12/20/2002	EXECUTION OF MODIFIED CHENEY SCANNING IN A MULTITHREADED PROCESSING ENVIRONMENT	HUDSON, RICHARD L.
10094469	Not Issued	061	03/08/2002	SELF ATTACHING, NON-SLIPPING, NON-MARRING, LONG LASTING, LOW COST, REPLACEABLE, SOFT, UNIVERSAL VISE JAW COVERS	HUDSON, RICHARD C.
10037600	Not Issued	041	11/09/2001	READY TO WIRE TERMINAL ASSEMBLY WITH VIBRATION RESISTANT CLAMPING SCREWS	HUDSON, RICHARD
09990978	Not	030	11/20/2001	REMOVABLE OCCLUSION	HUDSON, RICHARD

	Issued			SYSTEM FOR ANEURYSM NECK	
09977509 709/248 Kakali Chaki	Not Issued	030	10/12/2001	HARDWARE TO SUPPORT NON-BLOCKING SYNCHRONIZATION	HUDSON, RICHARD L.
09900336	Not Issued	041	07/05/2001	METHODS FOR PROTECTION OF STRATIFIED SQUAMOUS EPITHELIUM AGAINST INJURY BY NOXIOUS SUBSTANCES AND NOVEL AGENTS FOR USE THEREFOR	HUDSON, RICHARD A.
09886068 711/137 Thanh Ng	Not Issued	030	06/20/2001	METHOD FOR USING CACHE PREFETCH FEATURE TO IMPROVE GARBAGE COLLECTION ALGORITHM	HUDSON, RICHARD L.
09885745 Ule	Not Issued	030	06/19/2001	METHOD FOR USING NON-TEMPORAL STORES TO IMPROVE GARBAGE COLLECTION ALGORITHM	HUDSON, RICHARD L.
09708744	Not Issued	161	11/08/2000	SMART BATTERY SYSTEM MANAGEMENT AND INTERFACE	HUDSON, RICHARD A.
09707476	6388447	150	11/07/2000	METHOD AND APPARATUS FOR BATTERY FUEL GAUGING	HUDSON, RICHARD A.
09450870	6324339	150	11/29/1999	BATTERY PACK INCLUDING INPUT AND OUTPUT WAVEFORM MODIFICATION CAPABILITY	HUDSON , RICHARD A.
09428318	6349466	150	10/28/1999	READY TO WIRE TERMINAL ASSEMBLY WITH VIBRATION RESISTANT CLAMPING SCREWS	HUDSON, RICHARD
09421430 709/100 George Obie	Not Issued	071	10/19/1999	METHOD FOR PRACTICAL CONCURRENT COPYING GARBAGE COLLECTION OFFERING MINIMAL THREAD BLOCK TIMES	HUDSON , RICHARD L.
09301084	6344048	150	04/28/1999	REMOVABLE OCCLUSION SYSTEM FOR ANEURYSM NECK	HUDSON , RICHARD
09174509	6173350	150	10/16/1998	SYSTEM AND METHOD FOR WRITING DATA TO A SERIAL BUS FROM A SMART BATTERY	HUDSON , RICHARD A.
09039528	6023151	150	03/16/1998	METHOD AND DEVICE FOR ENHANCING SMART BATTERY PERFORMANCE	HUDSON , RICHARD
08891011	5928260	150	07/10/1997	REMOVABLE OCCLUSION SYSTEM FOR ANEURYSM NECK	HUDSON , RICHARD
08880083	5865122	150	06/20/1997	APPARATUS FOR ATTACHING BUFFER STOP TO RAILROAD TRACK	HUDSON , RICHARD C.
08826109	Not Issued	161	03/27/1997	READY-TO-WIRE CIRCUIT BREAKER	HUDSON, , RICHARD H.

08816658	5795232	250	03/13/1997	DRIVE ASSEMBLY COUPLER	HUDSON , RICHARD L.
06008706	4289274	150	02/01/1979	RAIL TRACK	HUDSON , RICHARD CARNE

Inventor Search Completed: No Records to Display.

Search Another: Inventor **Last Name** HUDSON **First Name** RICHARD Search

To go back use Back button on your browser toolbar.

Back to PALM | ASSIGNMENT | OASIS | Home page

PALM INTRANET

Day : Saturday
Date: 7/19/2003
Time: 10:52:09

Inventor Name Search Result

Your Search was:

Last Name = SUBRAMONEY
First Name = SREENIVAS

Application#	Patent#	Status	Date Filed	Title	Inventor Name
10411742 711	Not Issued	030	04/10/2003	METHOD, APPARATUS AND ARTICLE FOR LOCK MANAGEMENT	SUBRAMONEY, SREENIVAS
09886068 711	Not Issued	030	06/20/2001	METHOD FOR USING CACHE PREFETCH FEATURE TO IMPROVE GARBAGE COLLECTION ALGORITHM	SUBRAMONEY, SREENIVAS
09885745	Not Issued	030	06/19/2001	METHOD FOR USING NON-TEMPORAL STORES TO IMPROVE GARBAGE COLLECTION ALGORITHM	SUBRAMONEY, SREENIVAS

Inventor Search Completed: No Records to Display.

Search Another: Inventor Last Name: SUBRAMONEY First Name: SREENIVAS Search

To go back use Back button on your browser toolbar.

Back to PALM | ASSIGNMENT | OASIS | Home page

Terms used **garbage collection dynamic cache** Found **44** of **661,433**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The Digital Library

Results 1 - 20 of 44 Result page: 1 2 3 next

Relevance scale

1 Selector table indexing & sparse arrays
Karel Driesen
October 1993 **ACM SIGPLAN Notices , Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications**, Volume 28 Issue 10
Full text available: pdf(1.30 MB) Additional Information: full citation, references, citings, index terms

2 Compilation and run-time systems: DELI: a new run-time control point
Giuseppe Desoli, Nikolay Mateev, Evelyn Duesterwald, Paolo Faraboschi, Joseph A. Fisher
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references

The Dynamic Execution Layer Interface (DELI) offers the following unique capability: it provides fine-grain control over the execution of programs, by allowing its clients to observe and optionally manipulate every single instruction---at run time---just before it runs. DELI accomplishes this by opening up an interface to the layer between the execution of software and hardware. To avoid the slowdown, DELI caches a private copy of the executed code and always runs out of its own private cache.In ...

3 Caching: A self-managing data cache for edge-of-network web applications
Khalil Amiri, Sanghyun Park, Renu Tewari
November 2002 **Proceedings of the eleventh international conference on Information and knowledge management**
Full text available: pdf(340.44 KB) Additional Information: full citation, abstract, references, index terms

Database caching at proxy servers enables dynamic content to be generated at the edge of the network, thereby improving the scalability and response time of web applications. The scale of deployment of edge servers coupled with the rising costs of their administration demand that such caching middleware be adaptive and self-managing. To achieve this, a cache must be dynamically populated and pruned based on the application query stream and access pattern. In this paper, we describe such a cache ...

Keywords: dynamic content, e-commerce, semantic caching

4 Non-volatile memory for fast, reliable file systems
Mary Baker, Satoshi Asami, Etienne Deprit, John Ouseterhout, Margo Seltzer
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international**

conference on Architectural support for programming languages and operating systems, Volume 27 Issue 9
Full text available: pdf(1.47 MB) Additional Information: full citation, references, citings, index terms

5 Optimizing method search with lookup caches and incremental coloring
Pascal André, Jean-Claude Royer
October 1992 **ACM SIGPLAN Notices , conference proceedings on Object-oriented programming systems, languages, and applications**, Volume 27 Issue 10
Full text available: pdf(1.70 MB) Additional Information: full citation, references, citings, index terms

Keywords: Smalltalk-80, coloring, efficiency, lookup caches, method search, object-oriented languages with classes, statistics

6 Session S4.1: power in memory and network processors: An integrated approach to reducing power dissipation in memory hierarchies
Jayaprakash Pisharath, Alok Choudhary
October 2002 **Proceedings of the international conference on Compilers, architecture, and synthesis for embedded systems**
Full text available: pdf(295.32 KB) Additional Information: full citation, abstract, references, index terms

In recent years, both performance and power have become key factors in efficient memory design. In this paper, we propose a systematic approach to reduce the energy consumption of the entire memory hierarchy. We first evaluate an existing power-aware memory system where memory modules can exist in different power modes, and then propose on-chip memory module buffers, called Energy-Saver Buffers (ESB), which reside in-between the L2 cache and main memory. ESBs reduce the additional overhead incur ...

Keywords: RDRAM, dynamic cache, energy-delay product, energy-saver buffers (ESB), integrated approach, power

7 OMPI: optimizing MPI programs using partial evaluation
Hirotaka Ogawa, Satoshi Matsuoka
November 1996 **Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(138.70 KB) Additional Information: full citation, abstract, references, index terms

MPI is gaining acceptance as a standard for message-passing in high-performance computing, due to its powerful and flexible support of various communication styles. However, the complexity of its API poses significant software overhead, and as a result, applicability of MPI has been restricted to rather regular, coarse-grained computations. Our OMPI (Optimizing MPI) system removes much of the excess overhead by employing partial evaluation techniques, which exploit static information of MPI ...

8 Modeling Rate-Based Dynamic Cache Sharing for Distributed VOD Systems
B. Sonah, M. R. Ito
March 2000 **Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00)**
Full text available: Publisher Site Additional Information: full citation, abstract

A distributed VOD system includes several VOD subsystems, each VOD sub-system consisting of an archive server (AS), a continuous media server (CMS) and a medata DB. A VOD sub-system employs an object replacement algorithm by which a video is selected to be replaced by a new video. Upon a miss, the VOD system must decide onto which CMS to load the new video.In this paper, we address this issue by modeling a dynamic approach of

logically sharing the overall CMS cache space among the CMS's based on ...

9 A dynamic cache sub-block design to reduce false sharing
Murali Kadiyala, Laxmi N. Bhuyan
October 1995 **Proceedings of the 1995 International Conference on Computer Design: VLSI in Computers and Processors**
Full text available: Publisher Site Additional Information: full citation, abstract

Parallel applications differ from significant bus traffic due to the transfer of shared data. Large block sizes exploit locality and decrease the effective memory access time. It also has a tendency to group data together even though only a part of it is needed by any one processor. This is known as the false sharing problem. This research presents a dynamic sub-block coherence protocol which minimizes false sharing by trying to dynamically locate the point of false reference. Sharing traffic is ...

Keywords: bus traffic, cache storage, dynamic cache sub-block design, dynamic sub-block coherence protocol, false sharing, memory architecture, memory protocols, simulation results

10 Using dynamic cache management techniques to reduce energy in a high-performance processor
Nikolaos Bellas, Ibrahim Hajj, Constantine Polychronopoulos
August 1999 **Proceedings of the 1999 international symposium on Low power electronics and design**
Full text available: pdf(746.50 KB) Additional Information: full citation, citings, index terms

11 Dynamic cache partitioning for vector and matrix computations
Dana Mark Madsen
January 1997 Doctoral Thesis, Cornell University
Additional Information: full citation, index terms

Keywords: vector computations

12 Load execution latency reduction
Bryan Black, Brian Mueller, Stephanie Postal, Ryan Rakvic, Noppanunt Utamaphethai, John Paul Shen
July 1998 **Proceedings of the 12th international conference on Supercomputing**
Full text available: pdf(1.10 MB) Additional Information: full citation, references, citings, index terms

Keywords: load address prediction, load execution, load/store alias, speculative execution, value prediction

13 Page placement algorithms for large real-indexed caches
R. E. Kessler, Mark D. Hill
November 1992 **ACM Transactions on Computer Systems (TOCS)**, Volume 10 Issue 4
Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings, index terms

When a computer system supports both paged virtual memory and large real-indexed caches, cache performance depends in part on the main memory page placement. To date,

most operating systems place pages by selecting an arbitrary page frame from a pool of page frames that have been made available by the page replacement algorithm. We give a simple model that shows that this naive (arbitrary) page placement leads to up to 30% unnecessary cache conflicts. We develop several page placement algor ...

14 Routing I: Novel architectures for P2P applications: the continuous-discrete approach
Moni Naor, Udi Wieder
June 2003 **Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures**
Full text available: pdf(260.13 KB) Additional Information: full citation, abstract, references, index terms

We propose a new approach for constructing P2P networks based on a dynamic decomposition of a continuous space into cells corresponding to processors. We demonstrate the power of these design rules by suggesting two new architectures, one for DHT (Distributed Hash Table) and the other for dynamic expander networks. The DHT network, which we call Distance Halving allows logarithmic routing and load, while preserving constant degrees. It offers an optimal tradeoff between the degree and the dilati ...

Keywords: distributed systems, fault tolerance, hash tables, peer-to-peer

15 Pursuing the Performance Potential of Dynamic Cache Line Sizes
October 1999 **Proceedings of the 1999 IEEE International Conference on Computer Design**
Full text available: Publisher Site Additional Information: full citation, abstract

In this paper we examine the application of offline algorithms for determining the optimal sequence of loads and superloads (a load of multiple consecutive cache lines) for direct-mapped caches. We evaluate potential gains in terms of miss rate and bandwidth and find that in many cases optimal superloading can noticeably reduce the miss rate without appreciably increasing bandwidth. Then we examine how this performance potential might be realized. We examine the effectiveness of a dynamic online ...

Keywords: Cache Performance, Line Size, Optimal Algorithm, Prediction, Profiling

16 Dynamic Caching of Query Results for Decision Support Systems
Junho Shim, Peter Scheuermann, Radek Vingralek
July 1999 **Proceedings of the 11th International Conference on Scientific on Scientific and Statistical Database Management**
Full text available: Publisher Site Additional Information: full citation, abstract

The response time of DSS (Decision Support System) queries is typically several orders of magnitude higher than the response time of OLTP (OnLine Transaction Processing) queries. Since DSS queries are often submitted interactively, techniques for reducing their response time are becoming increasingly important. We argue that caching of query results is one such technique particularly well suited to the DSS environment. We have designed a query cache manager for such an environment. The cache man ...

17 Memory System Support for Dynamic Cache Line Assembly
Lixin Zhang, Venkata K. Pingali, Bharat Chandramouli, John B. Carter
November 2000 **Revised Papers from the Second International Workshop on Intelligent Memory Systems**
Additional Information: full citation

18 Using dynamic cache management techniques to reduce energy in general purpose processors

Nikolaos E. Bellas, Ibrahim N. Hajj, Constantine D. Polychronopoulos
December 2000 **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Volume 8 Issue 6
Additional Information: full citation, index terms

Keywords: low-power-design, memory, performance-trade-offs, system-level

19 Processor-based system: Tuning of loop cache architectures to programs in embedded system design
Susan Cotterell, Frank Vahid
October 2002 **Proceedings of the 15th international symposium on System Synthesis**
Full text available: pdf(58.95 KB) Additional Information: full citation, abstract, references, index terms

Adding a small loop cache to a microprocessor has been shown to reduce average instruction fetch energy for various sets of embedded system applications. With the advent of core-based design, embedded system designers can now tune a loop cache architecture to best match a specific application. We developed an automated simulation environment to find the best loop cache architecture for a given application and technology. Using this environment, we show significant variation in the best architect ...

Keywords: architecture tuning, cores, customized architectures, embedded systems, filter cache, instruction fetching, loop cache, low energy, low power, memory hierarchy, synthesis, tuning

20 An Optimal Cache for a Federated Database System
Alfredo Goñi, Arantza Illarramendi, Eduardo Mena, José Miguel Blanco
September 1997 **Journal of Intelligent Information Systems**, Volume 9 Issue 2
Full text available: Publisher Site Additional Information: full citation, abstract, index terms

Federated database systems allow users to query different autonomous databases with a single request. The answer to those requests must be found on the underlying databases. This answering process can be improved if some data are cached within the federated database system. The article presents an approach that allows the definition of an optimal cache for a federated database system according to a set of parameters. We show the types of objects to be cached, the cost model used to dec ...

Keywords: caching techniques, description logics, federated databases, query processing

Results 1 - 20 of 44 Result page: 1 2 3 next

Terms used **garbage collection dynamic cache** Found **44** of **661,433**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The Digital Library

Results 21 - 40 of 44 Result page: previous 1 2 3 next

Relevance scale

21 Session 5: Fine-grain CAM-tag cache resizing using miss tags

Michael Zhang, Krste Asanović
August 2002 **Proceedings of the 2002 international symposium on Low power electronics and design**

Full text available: pdf(220.91 KB) Additional Information: full citation, abstract, references, index terms

A new dynamic cache resizing scheme for low-power CAM-tag caches is introduced. A control algorithm that is only activated on cache misses uses a duplicate set of tags, the **miss tags**, to minimize active cache size while sustaining close to the same hit rate as a full size cache. The cache partitioning mechanism saves both switching and leakage energy in unused partitions with little impact on cycle time. Simulation results show that the scheme saves 28--56% of data cache energy and 34--49 ...

Keywords: cache resizing, content-addressable-memory, energy efficiency, leakage current, low-power

22 An accurate and efficient performance analysis technique for multiprocessor snooping cache-consistency protocols

M. K. Vernon, E. D. Lazowska, J. Zahorjan
May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture**, Volume 16 Issue 2

Full text available: pdf(999.88 KB) Additional Information: full citation, abstract, references, citings, index terms

A number of dynamic cache consistency protocols have been developed for multiprocessors having a shared bus interconnect between processors and shared memory. The relative performance of these protocols has been studied extensively using simulation and detailed analytical models based on Markov chain techniques. Both of these approaches use relatively detailed models, which capture cache and bus interference rather precisely, but which are highly expensive to evaluate. In this paper, we inv ...

23 Disk cache—miss ratio analysis and design considerations

Alan J. Smith
August 1985 **ACM Transactions on Computer Systems (TOCS)**, Volume 3 Issue 3

Full text available: pdf(3.13 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The current trend of computer system technology is toward CPUs with rapidly increasing processing power and toward disk drives of rapidly increasing density, but with disk performance increasing very slowly if at all. The implication of these trends is that at some point the processing power of computer systems will be limited by the throughput of the

input/output (I/O) system. A solution to this problem, which is described and evaluated in this paper, is disk cache

24 Predicting data cache misses in non-numeric applications through correlation profiling

Todd C. Mowry, Chi-Keung Luk

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(876.36 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

To maximize the benefit and minimize the overhead of software-based latency tolerance techniques, we would like to apply them precisely to the set of dynamic references that suffer cache misses. Unfortunately, the information provided by the state-of-the-art cache miss profiling technique (summary profiling) is inadequate for references with intermediate miss ratios - it results in either failing to hide latency, or else inserting unnecessary overhead. To overcome this problem, we propose and ev ...

Keywords: profiling, cache miss prediction, correlation, non-numeric applications, latency tolerance.

25 Compiler-directed cache polymorphism

J. S. Hu, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, H. Saputra, W. Zhang

June 2002 **ACM SIGPLAN Notices , Proceedings of the joint conference on Languages, compilers and tools for embedded systems: software and compilers for embedded systems**, Volume 37 Issue 7

Full text available: pdf(419.50 KB) Additional Information: full citation, abstract, references, index terms

Classical compiler optimizations assume a fixed cache architecture and modify the program to take best advantage of it. In some cases, this may not be the best strategy because each loop nest might work best with a different cache configuration and transforming a nest for a given fixed cache configuration may not be possible due to data dependences. Working with a fixed cache configuration can also increase energy consumption in loops where the best required configuration is smaller than the def ...

Keywords: cache locality, cache polymorphism, compilers, data reuse, embedded software, energy consumption

26 Power-and Energy-Aware Computing: Energy-efficient instruction cache using page-based placement

S. Kim, N. Vijaykrishnan, M. Kandemir, M. J. Irwin

November 2001 **Proceedings of the international conference on Compilers, architecture, and synthesis for embedded systems**

Full text available: pdf(185.58 KB) Additional Information: full citation, abstract, references, citings

Energy consumption is a crucial factor in designing battery-operated embedded and mobile systems. The memory system is a major contributor to the system energy in such environments. In order to optimize energy and energy-delay in the memory system, we investigate ways of splitting the instruction cache into several smaller units, each of which is a cache by itself (called *subcache*). The subcache architecture employs a page-based placement strategy, a dynamic cache line remapping policy an ...

27 Memory hierarchy reconfiguration for energy and performance in general-purpose processor architectures

Rajeev Balasubramonian, David Albonesi, Alper Buyuktosunoglu, Sandhya Dwarkadas

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(155.56 KB) ps(663.39 KB) Additional Information: full citation, references, citings, index terms

28 Using a knowledge cache for interactive discovery of association rules
Biswadeep Nag, Prasad M. Deshpande, David J. DeWitt
August 1999 **Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining**
Full text available: pdf(1.26 MB) Additional Information: full citation, references, citings, index terms

29 Closing the window of vulnerability in multiphase memory transactions
John Kubiatowicz, David Chaiken, Anant Agarwal
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9
Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

Multiprocessor architects have begun to explore several mechanisms such as prefetching, context-switching and software-assisted dynamic cache-coherence, which transform single-phase memory transactions in conventional memory systems into multiphase operations. Multiphase operations introduce a window of vulnerability in which data can be invalidated before it is used. Losing data due to invalidations introduces damaging livelock situations. This paper discusses the origins ...

30 Caches versus object allocation
J. Liedtke
October 1996 **Proceedings of the 5th International Workshop on Object Orientation in Operating Systems (IWOOOS '96)**
Full text available: Publisher Site Additional Information: full citation, abstract

Dynamic object allocation usually stresses the randomness of data memory usage; the variables of a dynamic cache working set are to some degree distributed stochastically in the virtual or physical address space. This interferes with cache architectures, since, currently, most of them are highly sensitive to access patterns. In the above mentioned stochastically distributed case, the true capacity is far below the cache size and largely differs from processor to processor. As a consequence, obje ...

Keywords: access patterns, cache architectures, data memory usage, dynamic cache working set, dynamic object allocation, memory management techniques, object allocation schemes, object-oriented programming, physical address space

31 On Request Forwarding for Dynamic Web Caching Hierarchies
Cho-Yu Chiang, Yingjie Li, Ming T. Liu, Mervin E. Muller
April 2000 **Proceedings of the The 20th International Conference on Distributed Computing Systems (ICDCS 2000)**
Full text available: Publisher Site Additional Information: full citation, abstract

Based on Caching Neighborhood Protocol (CNP), we proposed a Web caching scheme featuring dynamic caching hierarchies as its underlying infrastructure [1]. Dynamic Web caching hierarchies consist of proxy servers building hierarchies on a per request basis, in contrast to static Web caching hierarchies that comprise proxy servers preconfigured into hierarchies. Concerns on overheads and efficiency in forwarding requests individually drove conventional Web caching schemes to use static Web caching ...

32 Morphable Cache Architectures: Potential Benefits
I. Kadayif, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, J. Ramanujam
August 2001 **ACM SIGPLAN Notices**, Volume 36 Issue 8

Full text available: pdf(302.99 KB) Additional Information: full citation, abstract, references, citings, index terms

Computer architects have tried to mitigate the consequences of high memory latencies using a variety techniques. An example of these techniques is multi-level caches to counteract the latency that results from having a memory that is slower than the processor. Recent research has demonstrated that compiler optimizations that modify data layouts and restructure computation can be successful in improving memory system performance. However, in many cases, working with a fixed cache configuration ...

33 Dynamic services and analysis: Engineering and hosting adaptive freshness-sensitive web applications on data centers

Wen-Syan Li, Oliver Po, Wang-Pin Hsiung, K. Selçuk Candan, Divyakant Agrawal
May 2003 **Proceedings of the twelfth international conference on World Wide Web**

Full text available: pdf(10.31 MB) Additional Information: full citation, abstract, references, index terms

Wide-area database replication technologies and the availability of content delivery networks allow Web applications to be hosted and served from powerful data centers. This form of application support requires a complete Web application suite to be distributed along with the database replicas. A major advantage of this approach is that dynamic content is served from locations closer to users, leading into reduced network latency and fast response times. However, this is achieved at the expense ...

Keywords: database-driven web applications, dynamic content, freshness, response time, net-work latency, web acceleration

34 Synthesis of customized loop caches for core-based embedded systems

Susan Cotterell, Frank Vahid
November 2002 **Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(92.19 KB) Additional Information: full citation, abstract, references

Embedded system programs tend to spend much time in small loops. Introducing a very small loop cache into the instruction memory hierarchy has thus been shown to substantially reduce instruction fetch energy. However, loop caches come in many sizes and variations -- using the configuration best on the average may actually result in worsened energy for a specific program. We therefore introduce a loop cache exploration tool that analyzes a particular program's profile, rapidly explores the possib ...

Keywords: architecture tuning, customized architectures, embedded systems, estimation, instruction fetching, loop cache, low energy, low power, memory hierarchy, synthesis, tuning

35 A locality-preserving cache-oblivious dynamic dictionary

Michael A. Bender, Ziyang Duan, John Iacono, Jing Wu
January 2002 **Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract

This paper presents a simple dictionary structure designed for a hierarchical memory. The proposed data structure is *cache oblivious* and *locality preserving*. A cache-oblivious data structure has memory performance optimized for all levels of the memory hierarchy even though it has no memory-hierarchy-specific parameterization. A locality-preserving dictionary maintains elements of similar key values stored close together for fast access to ranges of data with consecutive keys.The d ...

36 Proxy-server architectures for OLAP

Panos Kalnis, Dimitris Papadias
May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2

Full text available: pdf(215.70 KB) Additional Information: full citation, abstract, references, citings, index terms

Data warehouses have been successfully employed for assisting decision making by offering a global view of the enterprise data and providing mechanisms for On-Line Analytical processing. Traditionally, data warehouses are utilized within the limits of an enterprise or organization. The growth of Internet and WWW however, has created new opportunities for data sharing among ad-hoc, geographically spanned and possibly mobile users. Since it is impractical for each enterprise to set up a worldwi ...

37 Cache investment: integrating query optimization and distributed data placement
Donald Kossmann, Michael J. Franklin, Gerhard Drasch, Wig Ag
December 2000 **ACM Transactions on Database Systems (TODS)**, Volume 25 Issue 4

Full text available: pdf(210.67 KB) Additional Information: full citation, abstract, references, citings, index terms

Emerging distributed query-processing systems support flexible execution strategies in which each query can be run using a combination of data shipping and query shipping. As in any distributed environment, these systems can obtain tremendous performance and availability benefits by employing dynamic data caching. When flexible execution and dynamic caching are combined, however, a circular dependency arises: Caching occurs as a by-product of query operator placement, but query operator pl ...

Keywords: cache investment, caching, client-server database systems, data shipping, dynamic data placement, query optimization, query shipping

38 Application-specific memory management for embedded systems using software-controlled caches
Derek Chiou, Prabhat Jain, Larry Rudolph, Srinivas Devadas
June 2000 **Proceedings of the 37th conference on Design automation**

Full text available: pdf(76.30 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose a way to improve the performance of embedded processors running data-intensive applications by allowing software to allocate on-chip memory on an application-specific basis. On-chip memory in the form of cache can be made to act like scratch-pad memory via a novel hardware mechanism, which we call column caching. Column caching enables dynamic cache partitioning in software, by mapping data regions to a specified sets of cache "columns" or "ways ...

39 Linux Vs. Windows NT and OS/2
Bernie Thompson
January 1994 **Linux Journal**

Full text available: html(16.27 KB) Additional Information: full citation, abstract, index terms

We continue to see media blurbs and ads for both Microsoft's Windows NT and IBM's OS/2. Both promise to be the operating system that we need and to take advantage of the Intel 386 and beyond

40 Caching multidimensional queries using chunks
Prasad M. Deshpande, Karthikeyan Ramasamy, Amit Shukla, Jeffrey F. Naughton
June 1998 **ACM SIGMOD Record , Proceedings of the 1998 ACM SIGMOD international conference on Management of data**, Volume 27 Issue 2

Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings, index terms

Caching has been proposed (and implemented) by OLAP systems in order to reduce response times for multidimensional queries. Previous work on such caching has considered table level caching and query level caching. Table level caching is more suitable for static schemes. On the other hand, query level caching can be used in dynamic schemes, but is too coarse for "large" query results. Query level caching has the further drawback for small query results in that it is only effectiv ...

Results 21 - 40 of 44 Result page: previous 1 **2** 3 next

Terms used **garbage collection dynamic cache** Found **44** of **661,433**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The Digital Library

Results 41 - 44 of 44 Result page: previous 1 2 **3**

Relevance scale

41 Informing memory operations: memory performance feedback mechanisms and their applications

Mark Horowitz, Margaret Martonosi, Todd C. Mowry, Michael D. Smith
May 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 2

Full text available: pdf(344.74 KB) Additional Information: full citation, abstract, references, index terms, review

Memory latency is an important bottleneck in system performance that cannot be adequately solved by hardware alone. Several promising software techniques have been shown to address this problem successfully in specific situations. However, the generality of these software approaches has been limited because current architecturtes do not provide a fine-grained, low-overhead mechanism for observing and reacting to memory behavior directly. To fill this need, this article proposes a new class ...

Keywords: cache miss notification, memory latency, processor architecture

42 Adaptive data prefetching using cache information

Ando Ki, Alan E. Knowles
July 1997 **Proceedings of the 11th international conference on Supercomputing**

Full text available: pdf(1.89 MB) Additional Information: full citation, references, citings, index terms

43 Transactional client-server cache consistency: alternatives and performance

Michael J. Franklin, Michael J. Carey, Miron Livny
September 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 3

Full text available: pdf(452.41 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Client-server database systems based on a data shipping model can exploit client memory resources by caching copies of data items across transaction boundaries. Caching reduces the need to obtain data from servers or other sites on the network. In order to ensure that such caching does not result in the violation of transaction semantics, a transactional cache consistency maintenance algorithm is required. Many such algorithms have been proposed in the literature and, as all provide the sam ...

44 Stack caching for interpreters

M. Anton Ertl
June 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 conference on Programming language design and implementation**, Volume 30 Issue 6

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms

An interpreter can spend a significant part of its execution time on accessing arguments of virtual machine instructions. This paper explores two methods to reduce this overhead for virtual stack machines by caching top-of-stack values in (real machine) registers. The dynamic method is based on having, for every possible state of the cache, one specialized version of the whole interpreter; the execution of an instruction usually changes the state of the cache and the next i ...

Results 41 - 44 of 44 Result page: previous 1 2 **3**

Terms used **how to collect garbage without using a cache?** Found **94,787** of **661,433**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The Digital Library

too many hits marked only page 1

Results 1 - 20 of 200 Result page: 1 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Tuning garbage collection for reducing memory system energy in an embedded java environment

G. Chen, R. Shetty, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, M. Wolczko

November 2002 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 1 Issue 1

Full text available: pdf(740.23 KB) Additional Information: full citation, abstract, references, index terms

Java has been widely adopted as one of the software platforms for the seamless integration of diverse computing devices. Over the last year, there has been great momentum in adopting Java technology in devices such as cellphones, PDAs, and pagers where optimizing energy consumption is critical. Since, traditionally, the Java virtual machine (JVM), the cornerstone of Java technology, is tuned for performance, taking into account energy consumption requires reevaluation, and possibly redesign of t ...

Keywords: Garbage collector, Java Virtual Machine (JVM), K Virtual Machine (KVM), low power computing

2 Concurrent garbage collection using hardware-assisted profiling

Timothy H. Heil, James E. Smith

October 2000 **ACM SIGPLAN Notices , Proceedings of the second international symposium on Memory management**, Volume 36 Issue 1

Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, citings, index terms

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

3 Garbage collection for a client-server persistent object store

Laurent Amsaleg, Michael J. Franklin, Olivier Gruber

August 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 3

Full text available: pdf(267.18 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe an efficient server-based algorithm for garbage collecting persistent object stores in a client-server environmnet. The algorithm is incremental and runs concurrently with client transactions. Unlike previous algorithms, it does not hold any transactional locks on data and does non require callbacks to clients. It is fault-tolerant, but performs very little logging. The algorithm has been designed to be integrated into existing systems, and

therefore it works with standard i ...

Keywords: client-server system, logging, persistent object-store, recovery

4 The measured cost of copying garbage collection mechanisms

Michael W. Hicks, Jonathan T. Moore, Scott M. Nettles

August 1997 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN international conference on Functional programming**, Volume 32 Issue 8

Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references, index terms

We examine the costs and benefits of a variety of copying garbage collection (GC) mechanisms across multiple architectures and programming languages. Our study covers both low-level object representation and copying issues as well as the mechanisms needed to support more advanced techniques such as generational collection, large object spaces, and type segregated areas.Our experiments are made possible by a novel performance analysis tool, *Oscar*. Oscar allows us to capture snapshots of pr ...

5 Garbage collection in object-oriented databases using transactional cyclic reference counting

P. Roy, S. Seshadri, A. Silberschatz, S. Sudarshan, S. Ashwin

August 1998 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 7 Issue 3

Full text available: pdf(180.00 KB) Additional Information: full citation, abstract

Garbage collection is important in object-oriented databases to free the programmer from explicitly deallocating memory. In this paper, we present a garbage collection algorithm, called Transactional Cyclic Reference Counting (TCRC), for object-oriented databases. The algorithm is based on a variant of a reference-counting algorithm proposed for functional programming languages The algorithm keeps track of auxiliary reference count information to detect and collect cyclic garbage. The algorithm ...

6 Memory subsystem performance of programs using copying garbage collection

Amer Diwan, David Tarditi, Eliot Moss

February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available: pdf(1.28 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Heap allocation with copying garbage collection is believed to have poor memory subsystem performance. We conducted a study of the memory subsystem performance of heap allocation for memory subsystems found on many machines. We found that many machines support heap allocation poorly. However, with the appropriate memory subsystem organization, heap allocation can have good memory subsystem performance.

7 Cache performance of garbage-collected programs

Mark B. Reinhold

June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation**, Volume 29 Issue 6

Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms

As processor speeds continue to improve relative to main-memory access times, cache performance is becoming an increasingly important component of program performance. Prior work on the cache performance of garbage-collected programs either argues or assumes that conventional garbage-collection methods will yield poor performance, and has therefore concentrated on new collection algorithms designed specifically to improve cache-level reference locality. This paper argues to the c ...

8 Caching considerations for generational garbage collection

Paul R. Wilson, Michael S. Lam, Thomas G. Moher
January 1992 **ACM SIGPLAN Lisp Pointers , Proceedings of the 1992 ACM conference on LISP and functional programming**, Volume V Issue 1
Full text available: pdf(1.09 MB) Additional Information: full citation, references, citings, index terms

9 Concurrent compacting garbage collection of a persistent heap
James O'Toole, Scott Nettles, David Gifford
December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5
Full text available: pdf(1.50 MB) Additional Information: full citation, references, citings, index terms

10 Partitioned garbage collection of a large object store
Umesh Maheshwari, Barbara Liskov
June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data**, Volume 26 Issue 2
Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

We present new techniques for efficient garbage collection in a large persistent object store. The store is divided into partitions that are collected independently using information about inter-partition references. This information is maintained on disk so that it can be recovered after a crash. We use new techniques to organize and update this information while avoiding disk accesses. We also present a new global marking scheme to collect cyclic garbage across partitions. Global marking ...

Keywords: cyclic garbage, garbage collection, object database, partitions

11 Garbage collecting the Internet: a survey of distributed garbage collection
Saleh E. Abdullahi, Graem A. Ringwood
September 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 3
Full text available: pdf(337.65 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Internet programming languages such as Java present new challenges to garbage-collection design. The spectrum of garbage-collection schema for linked structures distributed over a network are reviewed here. Distributed garbage collectors are classified first because they evolved from single-address-space collectors. This taxonomy is used as a framework to explore distribution issues: locality of action, communication overhead and indeterministic communication latency.

Keywords: automatic storage reclamation, distributed, distributed file systems, distributed memories, distributed object-oriented management, memory management, network communication, object-oriented databases, reference counting

12 Very concurrent mark-&-sweep garbage collection without fine-grain synchronization
Lorenz Huelsbergen, Phil Winterbottom
October 1998 **ACM SIGPLAN N tices , Proceedings of the first international symposium on Memory management**, Volume 34 Issue 3
Full text available: pdf(1.36 MB) Additional Information: full citation, abstract, references, citings, index terms

We describe a new incremental algorithm for the concurrent reclamation of a program's allocated, yet unreachable, data. Our algorithm is a variant of mark-&-sweep collection that---unlike prior designs---runs mutator, marker, and sweeper threads concurrently

without explicit fine-grain synchronization on shared-memory multiprocessors. A global, but infrequent, synchronization coordinates the per-object coloring marks used by the three threads; fine-grain synchronization is achieve ...

13 Creating and preserving locality of java applications at allocation and garbage collection times

Yefim Shuf, Manish Gupta, Hubertus Franke, Andrew Appel, Jaswinder Pal Singh

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications**, Volume 37 Issue 11

Full text available: pdf(180.20 KB) Additional Information: full citation, abstract, references, index terms

The growing gap between processor and memory speeds is motivating the need for optimization strategies that improve data locality. A major challenge is to devise techniques suitable for pointer-intensive applications. This paper presents two techniques aimed at improving the memory behavior of pointer-intensive applications with dynamic memory allocation, such as those written in Java. First, we present an allocation time object placement technique based on the recently introduced notion of *p* ...

Keywords: *JVM, Java, garbage collection, heap traversal, locality, locality based graph traversal, memory allocation, memory management, object co-allocation, object placement, prolific types, run-time systems*

14 Contaminated garbage collection

Dante J. Cannarozzi, Michael P. Plezbert, Ron K. Cytron

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Full text available: pdf(559.20 KB) Additional Information: full citation, abstract, references, citings, index terms

We describe a new method for determining when an object can be garbage collected. The method does not require marking live objects. Instead, each object X is dynamically associated with a stack frame M, such that Xis collectable when M pops. Because X could have been dead earlier, our method is conservative. Our results demonstrate that the method nonetheless identifies a large percentag ...

15 Concurrent replicating garbage collection

James O'Toole, Scott Nettles

July 1994 **ACM SIGPLAN Lisp Pointers , Proceedings of the 1994 ACM conference on LISP and functional programming**, Volume VII Issue 3

Full text available: pdf(919.87 KB) Additional Information: full citation, abstract, references, citings, index terms

We have implemented a concurrent copying garbage collector that uses replicating garbage collection. In our design, the client can continuously access the heap during garbage collection. No low-level synchronization between the client and the garbage collector is required on individual object operations. The garbage collector replicates live heap objects and periodically synchronizes with the client to obtain the client's current root set and mutation log. An experimental implementation usi ...

16 Reducing garbage collector cache misses

Hans-J. Boehm

October 2000 **ACM SIGPLAN Notices , Proceedings of the second international symposium n Memory management**, Volume 36 Issue 1

Full text available: pdf(774.19 KB) Additional Information: full citation, abstract, citings, index terms

Cache misses are currently a major factor in the cost of garbage collection, and we expect them to dominate in the future. Traditional garbage collection algorithms exhibit relatively little temporal locality; each live object in the heap is likely to be touched exactly once

during each garbage collection. We measure two techniques for dealing with this issue: prefetch-on-grey, and lazy sweeping. The first of these is new in this context. Lazy sweeping has been in common use for a decade. It ...

17 Portable, unobtrusive garbage collection for multiprocessor systems

Damien Doligez, Georges Gonthier

February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

We describe and prove the correctness of a new concurrent mark-and-sweep garbage collection algorithm. This algorithm derives from the classical on-the-fly algorithm from Dijkstra et al. [9]. A distinguishing feature of our algorithm is that it supports multiprocessor environments where the registers of running processes are not readily accessible, without imposing any overhead on the elementary operations of loading a register or reading or initializing a field. Furthermor ...

18 Concurrent garbage collection using program slices on multithreaded processors

Manoj Plakal, Charles N. Fischer

October 2000 **ACM SIGPLAN Notices , Proceedings of the second international symposium on Memory management**, Volume 36 Issue 1

Full text available: pdf(957.62 KB) Additional Information: full citation, abstract, citings, index terms

We investigate reference counting in the context of a multi-threaded architecture by exploiting two observations: (1) reference-counting can be performed by a transformed program slice of the mutator that isolates heap references, and (2) hardware trends indicate that microprocessors in the near future will be able to execute multiple concurrent threads on a single chip. We generate a reference-counting collector as a transformed program slice of an application and then execute this slice in ...

19 Fast out-of-order processor simulation using memoization

Eric Schnarr, James R. Larus

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 32 , 33 Issue 5 , 11

Full text available: pdf(1.43 MB) Additional Information: full citation, abstract, references, citings, index terms

Our new out-of-order processor simulatol; FastSim, uses two innovations to speed up simulation 8--15 times (vs. Wisconsin SimpleScalar) with no loss in simulation accuracy. First, FastSim uses speculative direct-execution to accelerate the functional emulation of speculatively executed program code. Second, it uses a variation on memoization---a well-known technique in programming language implementation---to cache microarchitecture states and the resulting simulator actions, and then "fast forw ...

Keywords: direct-execution, memoization, out-of-order processor simulation

20 Cache performance of fast-allocating programs

Marcelo J. R. Gonçalves, Andrew W. Appel

October 1995 **Proceedings of the seventh international conference on Functional programming languages and computer architecture**

Full text available: pdf(1.47 MB) Additional Information: full citation, references, citings, index terms

Terms used **moving garbage collection** Found **1** of **661,433**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The Digital Library

Results 1 - 1 of 1

Relevance scale

1 A non-fragmenting non-moving, garbage collector
Gustavo Rodriguez-Rivera, Michael Spertus, Charles Fiterman
October 1998 **ACM SIGPLAN Notices , Proceedings of the first international symposium on Memory management**, Volume 34 Issue 3
Full text available: pdf(750.52 KB) Additional Information: full citation, abstract, references, index terms

One of the biggest disadvantages of non-moving collectors compared to moving collectors has been their limited ability to deal with memory fragmentation. In this paper, we describe two techniques to reduce fragmentation without the need for moving live data. The first technique reduces internal fragmentation in BiBoP (Big-Bag-of-Pages) like allocators. The second technique reduces external fragmentation using virtual memory culls available in most modern operating systems. It can also reduce the ...

Keywords: conservative garbage collection, fragmentation, garbage collection, memory allocation, non-copying garbage collection

Results 1 - 1 of 1

Your search matched **4** of **950522** documents.

A maximum of **4** results are displayed, **25** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.
Then click **Search Again**.

garbage collection and moving Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

---

1 **Hardware and operating system support for conservative garbage collection**
Boehm, H.-J.;
Object Orientation in Operating Systems, 1991. Proceedings., 1991 International Workshop on , 17-18 Oct. 1991
Page(s): 61 -67

[Abstract] [PDF Full-Text (560 KB)] IEEE CNF

---

2 **Long-range dependence and self-similarity in World Wide Web proxy cache references**
Holt, A.;
Communications, IEE Proceedings- , Volume: 147 Issue: 6 , Dec. 2000
Page(s): 317 -321

[Abstract] [PDF Full-Text (420 KB)] IEE JNL

---

3 **Distributed/concurrent garbage collection in distributed shared mem ry systems**
Kordale, R.; Ahamad, M.; Shilling, J.;
Object Orientation in Operating Systems, 1993., Proceedings of the Third Internationa Workshop on , 9-10 Dec. 1993
Page(s): 51 -60

[Abstract] [PDF Full-Text (792 KB)] IEEE CNF

---

4 **M saic: a n n-intrusive c mplete garbage c llect r f r DSM systems**
Munro, D.S.; Falkner, K.E.; Lowry, M.C.; Vaughan, F.A.;
Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on , 15-18 May 2001

Page(s): 539 -546

[Abstract] [PDF Full-Text (716 KB)] **IEEE CNF**